EnMonitor: Experimentation over Large-scale Semantically Annotated Federated IoT data environment

Rachit Agarwal
Inria Paris, France
rachit.agarwal@inria.fr

David Gomez*
University of Cantabria, Spain
dgomez@tlmat.unican.es

Jorge Lanza
University of Cantabria, Spain
jlanza@tlmat.unican.es

Luis Sanchez
University of Cantabria, Spain
lsanchez@tlmat.unican.es

Nikolaos Georgantas
Inria Paris, France
nikolaos.georgantas@inria.fr

Valerie Issarny
Inria Paris, France
valerie.issarny@inria.fr

ABSTRACT

IoT applications are usually built on top of proprietary platforms that collect data from IoT devices. Furthermore, most applications rely on proprietary datasets, coming from their own sources. Thereby needing to deal with issues like interoperability and heterogeneity in the data. A solution is to add another layer (a platform in the middle that addresses the above-mentioned issues) and then build applications that use data made available via such platforms. As a proof of concept, we present `EnMonitor`, a prototype application, that is built on top of one such platform, called FIESTA-IoT. The application provides citizens with an understanding of the environment they live in with both local and global surrounding view.

KEYWORDS

Semantics, IoT, Citizen-Centric Application

ACM Reference Format:
Rachit Agarwal, David Gomez, Jorge Lanza, Luis Sanchez, Nikolaos Georgantas, and Valerie Issarny. 2018. EnMonitor: Experimentation over Large-scale Semantically Annotated Federated IoT data environment. In *Proceedings of The Web Conference (WWW2018).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Citizen-centric applications are focused on their well-being of citizens and provide information about their environment [5, 8]. These applications get information from dedicated platforms that collect data from available sensors using various sensing mechanisms (static, mobile or participatory). The use of such platforms limits a citizen from knowing much fine-grain information about the environment, as the applications are limited to receive the data from sensors attached to the specific platform. Using data from various other platforms as well across the domain of interests provides much more comprehensive knowledge about the environment. However, an application would need to deal with data coming from heterogeneous platforms. In order to mitigate such interoperability issues, much attention has been paid recently on the creation of a unified ontology that data-sources (also called `testbeds`) must comply in order to provide a common data model [1, 2]. Moreover, to ease development effort, a platform in a middle is introduced that is responsible for aggregating data made available via testbeds. Thus, a developer now has to only use the APIs of a middle platform to get the needed data.

The `FIESTA-IoT`[1] platform [3, 6] is one such EU H2020 initiative that enables federation of testbeds to help citizen-centric application developers provide a comprehensive large-scale fine-grain view. Note that, there are other EU H2020 and research initiatives [7] that talk about the federation of data or provide resources to build such applications; however, they lack interoperability, common understanding about the data and have a static snapshot of data. The data in the testbeds associated with FIESTA-IoT platform constantly evolve over time.

We utilize the FIESTA-IoT platform to build and demonstrate an application called `EnMonitor`[2] that showcases the power of the FIESTA-IoT platform and provides citizens near real-time comprehensive information about environmental conditions at different spatio-temporal scales. EnMonitor focuses on the environmental realm, displaying information about a number of different physical phenomena (e.g. temperature, noise level, relative humidity, solar radiation, the concentration of harmful particles in the air, etc.) that have health-related issues on citizens. As an example, using EnMonitor, a user could get information about surrounding environment before traveling or taking a certain path. Not limiting to this, EnMonitor also helps decision making bodies to make policies based on the comprehensive view.

EnMonitor has a threefold objective: *(a)* to provide near real-time information about the environment all around the globe, *(b)* to display it in an intuitive manner and *(c)* showcase the power of the FIESTA-IoT platform. Through an easy-to-use web-based graphical interface, users can pinpoint concrete regions on a map, select among different environmental phenomena, view different metrics (eg. heatmap) as shown in Fig. 1. Once the application is configured, EnMonitor periodically sends requests to the FIESTA-IoT platform for the results.

Thus, the value of EnMonitor is to not only offer citizens a holistic view of the environment around them but also to enable

*Also with ATOS Research and Innovation, Santander, Spain.

[1]https://platform.fiesta-iot.eu/portalui/
[2]http://fiesta-iot.tlmat-unican.es/enmonitor-demo

This paper is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution. In case of republication, reuse, etc., the following attribution should be used: "Published in WWW2018 Proceedings © 2018 International World Wide Web Conference Committee, published under Creative Commons CC BY 4.0 License."
WWW2018, April 2018, Lyon, France
© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

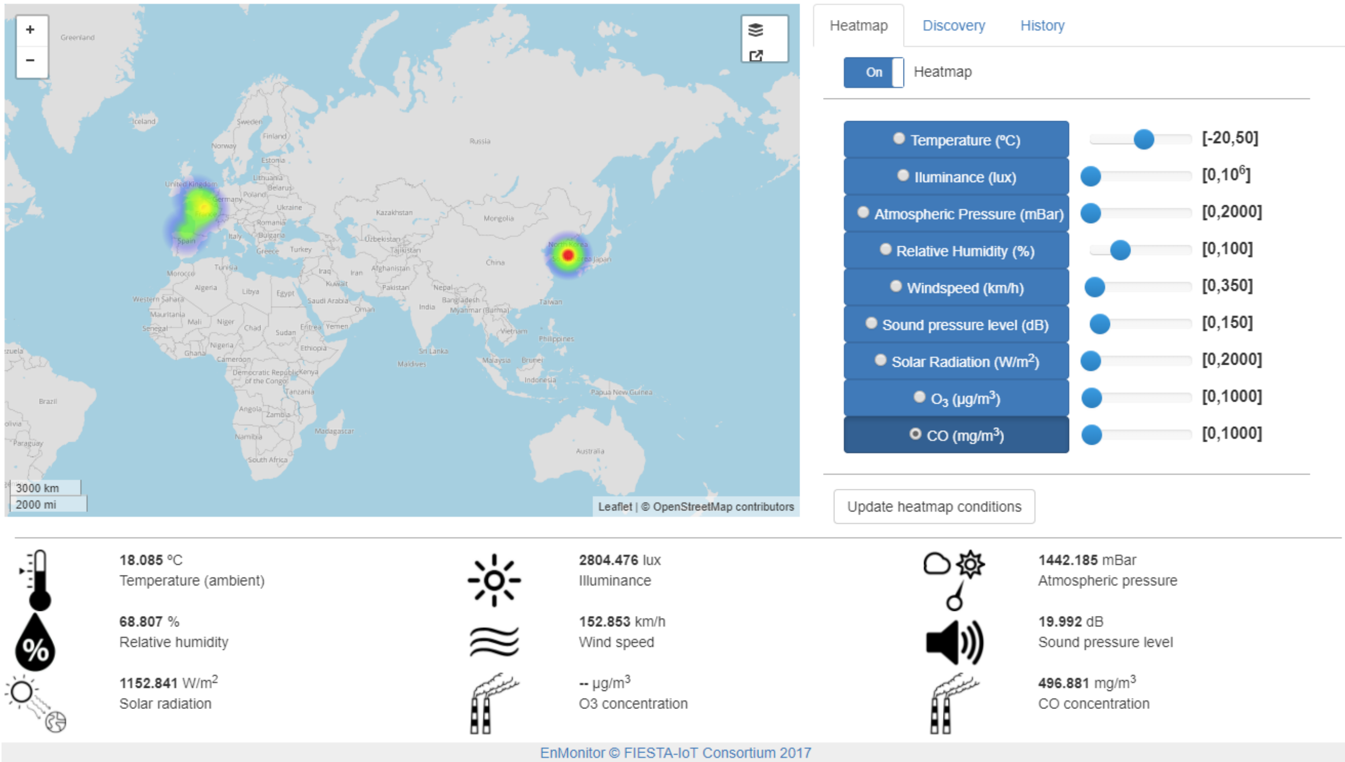


Figure 1: EnMonitor User Interface sample

policymakers to take advantage of such application to complement their legacy instrumentation tools and validate experimentation over large-scale semantically annotated federated IoT datastore.

In the following sections, we present EnMonitor and its interactions with the FIESTA-IoT platform. Following the description of EnMonitor, we present how EnMonitor can be reused.

2 SYSTEM OVERVIEW

In order to have reduced computational and networking overhead, we have designed EnMonitor with a client-server approach, as shown in Fig. 2. At the client side, only simple operations are carried out (basically, graphical input/output and lightweight configuration), while the server interacts with the FIESTA-IoT platform by means of SPARQL queries so as to extract information, such as *(a)* the resources that are registered within the semantic store of FIESTA-IoT platform (we refer to it as **resource discovery**. We define a resource as a device that is composed of one or many sensors.); *(b)* the observations measured by the sensors associated to those resources (we also call it as **observation harvest**). Besides, the server also caches the resultsets obtained from the required SPARQL queries and save them locally. This reduces the traffic between FIESTA-IoT platform and EnMonitor in a multi-user scenario, where most of the requests would be redundant. Additionally, for the sake of keeping track of what is happening in the application, we include a fully-fledged logging system that collects all the relevant events (e.g. errors and warnings) that occur between the components of the system.

The FIESTA-IoT platform allows federation of testbeds by enforcing semantics. It is supported by various components that provides a number of features, such as security (i.e. authentication & authorization), data storage and management services and an engine for the seamless execution of experiments, among other modules [3]. Due to space restrictions, we do not focus on all the interactions happening within the FIESTA-IoT platform; on the other hand, we focus on pointing out some key aspects that are necessary for EnMonitor and on how it interacts with the FIESTA-IoT platform.

One of the most important components of the FIESTA-IoT platform is the `IoT-Registry`. It enables the interactions between FIESTA-IoT components and distributed FIESTA-IoT ontology compliant semantic testbeds. Technically speaking, IoT-Registry is a Jena[3] based Semantic store that exposes a set of clearly documented both private and public REST APIs[4]. These APIs are implemented using RESTEasy[5] APIs, secured and can only be accessed if authentication and authorization step passes (this is managed by an instance of OpenAM platform). The APIs follow best practices and standards described in the WoT (Web of Things) literature.

Out of the available set of APIs, the most useful API to us is the SPARQL endpoint that brings about the possibility of querying the FIESTA-IoT semantic store (IoT-Registry). We refer the readers

[3] https://jena.apache.org
[4] https://platform.fiesta-iot.eu/iot-registry/docs/api.html
[5] http://resteasy.jboss.org

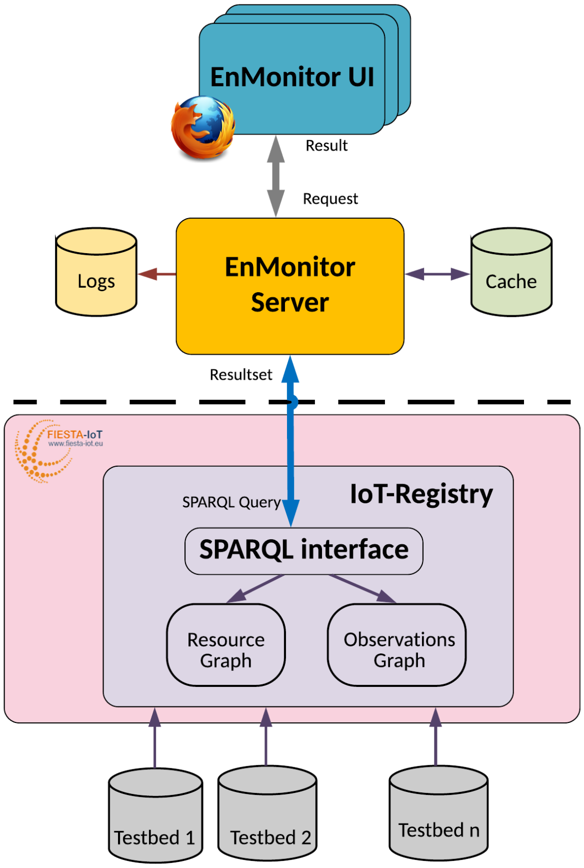


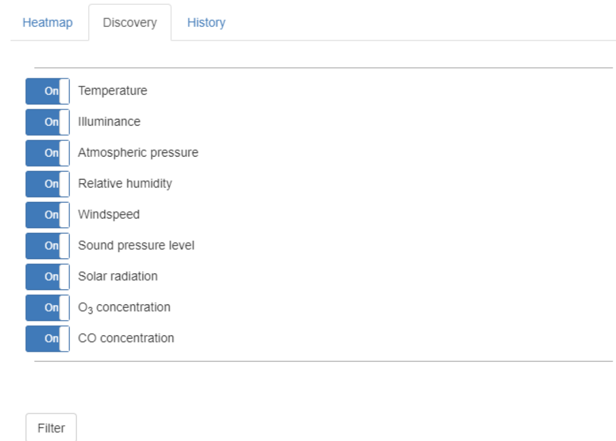
Figure 2: Application overview

to [4] to give a glimpse of the semantically annotated sensor data made available by this API. From the list, it can be seen that most of the data made available by the federated testbeds within the FIESTA-IoT realm relate to phenomena (also referred as quantity-kinds) associated to the environmental domain, that spans either marine, agriculture, smart-city or smart building scenarios. Within FIESTA-IoT, the available semantically annotated dataset is divided into two parts: `resource graph` (contains a description of the available sensors) and `observation graph` (contains observations made by the sensors in the resource graph). Moreover, a virtual `global graph` is also created within IoT-Registry that utilizes both resource and observation graph to enable generic queries. EnMonitor exploits both these graphs depending on the chosen scenario.

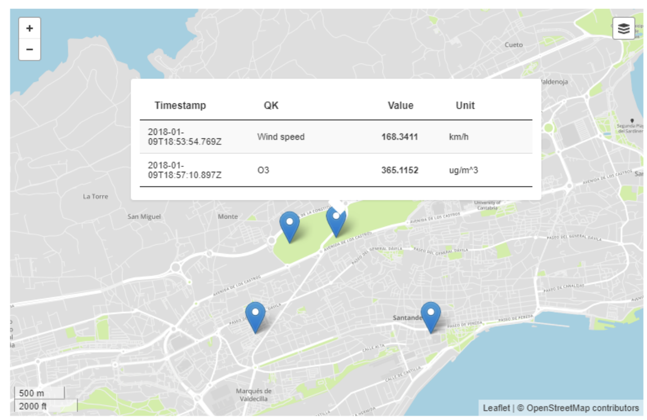
The EnMonitor UI, shown in Fig. 1 is implemented using JavaScript and HTML. It further uses popular JavaScript based libraries such as Leaflet[6], and D3[7] for visualization. The UI provides its users various features including:

- **Sensor/Resource discovery**. Users can discover sensors available/registered in a given area. Beside this area based filtering, they are also able to select and display only those

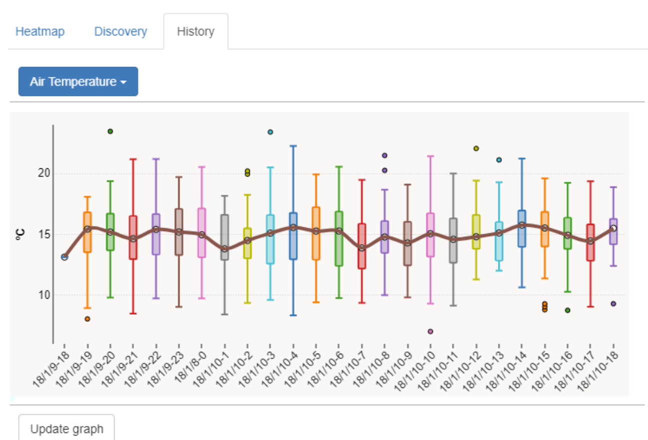
[6]http://leafletjs.com
[7]https://d3js.org



(a) Discovery of resources based on physical phenomena (quantity-kinds)



(b) Representation of the last observation collected by the sensors associated to a selected device/resource



(c) Graphical representation of historic values (only the ones that are displayed on the map)

Figure 3: Graphical Features

resources that are the owner of at least one sensor that match with the enabled quantity-kind toggles (shown in Fig. 3a. In the example, only the resources with combination of temperature, atmospheric pressure, relative humidity and wind-speed sensors will appear on the map).

- **Sensor data/Observation harvest**. Users can click on a resource to get the last measurements observed by each of its underlying sensors (recall that a resource can embrace one or more sensors), as captured in Fig. 3b. Thanks to the interoperability supported by the FIESTA-IoT platform, datasets from different testbeds have the same format and are fully compliant with the FIESTA-IoT ontology[8].
- **Heatmap**. Not limited to the individual and most recent observations, a user can also visualize the gradients perceived in a global perspective. Such visualizations enable detection of behavioral patterns that might be the consequence of strange behaviors (e.g. a fire in a forest would bring about drastic changes in temperature and relative humidity in the area). A user has to "switch on" the heatmap toggle (see Fig. 1) to activate the heatmap. If the toggle is disabled, instead of the heatmap, the resources are represented as markers on the map. Depending on the zoom level, these markers are clustered in order to facilitate the visualization.
- **Weather station**. Even though a significant number of measurements are gathered by each sensor, aggregation of data conveys a higher level of information. Thus, EnMonitor provides the average values of all the last observations captured within the area shown by the map (disregarding the phenomena-based filters). Note that, currently the weather station cannot be configured by the users. The weather station is depicted in the bottom frame of Fig. 1.
- **Historical data stats**. The scope of EnMonitor is not limited to just spatial features but extends to the spatio-temporal features as well. EnMonitor provides time-series based visualization for prediction of events that could be alarming and need immediate counteraction. Fig. 3c shows a 24-hour window of the air (ambient) temperature variation for a selected area. Future revisions of EnMonitor will allow modifications to (stretch/squeeze) the visualization interval (i.e. the x-axis). As an added functionality, a user can further modify the air temperature quantity-kind from the drop-down box. Upon this modification, the user has to "Update graph" to update the series.

A user can select any of the above features and customize them based on the needs. This triggers an interaction between the EnMonitor UI and the EnMonitor Server (implemented using NodeJS[9]). The communication channel between the server and the client is based on WebSockets, namely Socket.io[10], that enables a bi-directional real-time reliable link between the server and the client(s).

The server is in charge of contacting the FIESTA-IoT platform, receiving the semantically annotated resultsets from the FIESTA-IoT platform, performing basic analysis on the acquired resultsets, caching resultsets and maintaining application-level logs. The server specifically calls the `/queries/execute/resources` or the `/queries/execute/global` API of the IoT-Registry to execute the discovery or the observation harvest based SPARQL queries, respectively. Besides the call to these APIs, the only other API call that the server makes is to the Authentication API of the FIESTA-IoT platform (`/openam/json/authenticate`) to retrieve the access token.

3 SETUP AND REQUIREMENTS

For the demo purpose, it is not required to install the EnMonitor server as it is already configured by us and executes on a dedicated VM. Nevertheless, in case anybody wants to install and reuse the EnMonitor, its code is available[11] for download with a clear installation and requirement guide. Moreover, the EnMonitor UI is best viewed in Google Chrome browser installed on a machine that has access to high-speed Internet connection.

4 CONCLUSION

EnMonitor[12] presents a more fine-grained analysis of the environment to the citizen, thereby enabling them as well as policymakers to gain health benefits. EnMonitor, on one hand, provides different visualizations and metrics to understand the environment, and on another, shows the capabilities of the FIESTA-IoT platform. As future extensions, many aspects are planned to be integrated with EnMonitor. These include more near real-time statistical inferences, notifications to logged-in users and more sophisticated querying.

ACKNOWLEDGMENTS

This work is partially funded by the European project Federated Interoperable Semantic IoT/cloud Testbeds and Applications (FIESTA-IoT) from the European Union's Horizon 2020 Programme with the Grant Agreement No. CNECT-ICT-643943. The authors would like to thank the FIESTA-IoT consortium for the fruitful discussions.

REFERENCES

[1] Rachit Agarwal, David Gomez, Tarek Elsaleh, Amelie Gyrard, Jorge Lanza, Luis Sanchez, Nikolaos Georgantas, and Valerie Issarny. 2016. Unified IoT Ontology to Enable Interoperability and Federation of Testbeds. In *3rd IEEE World Forum on Internet of Things*. 70–75.

[2] Garvita Bajaj, Rachit Agarwal, Pushpendra Singh, Nikolaos Georgantas, and Valerie Issarny. 2017. A Study of Existing Ontologies in IoT-domain. *arXiv* (2017).

[3] Francois Carrez, David Gomez, Luis Sanchez, Jorge Lanza, Paul Grace, and Tarek Elsaleh. 2017. A Reference Architecture for Federating IoT Infrastructures Supporting Semantic Interoperability. In *2017 European Conference on Networks and Communications (EuCNC)*. 1–6.

[4] FIESTA-IoT. 2017. FIESTA-IoT Available Testdeds. http://fiesta-iot.eu/index.php/fiesta-testbeds/. (2017). Online; accessed 4 January 2018.

[5] Valerie Issarny, Vivien Mallet, Kinh Nguyen, PG Raverdy, Fadwa Rebhi, and Raphael Ventura. 2016. Do's and Don'ts in Mobile Phone Sensing Middleware: Learning from a Large-Scale Experiment. In *ACM/IFIP/ USENIX Middleware*. Trento.

[6] Jorge Lanza, Luis Sanchez, David Gomez, Tarek Elsaleh, Ronald Steinke, and Flavio Cirillo. 2016. A Proof of Concept for Semantically Interoperable Federation of IoT Experimentation Facilities. *Sensors* 16, 7 (June 2016), 1006.

[7] Ricardo Lopes Pereira, Pedro Cruz Sousa, Ricardo Barata, André Oliveira, and Geert" Monsieur. 2015. CitySDK Tourism API - building value around open data. *Journal of Internet Services and Applications* 6, 1 (2015), 24.

[8] SmartSantander. 2014. SmartSantander Maps portal. http://maps.smartsantander.eu/. (2014). Online; accessed 4 January 2018.

[8] http://ontology.fiesta-iot.eu/ontologyDocs/fiesta-iot.html

[9] https://nodejs.org/en/

[10] https://socket.io/

[11] https://github.com/fiesta-iot/enmonitor-demo

[12] Supplementary demo video is available at https://youtu.be/kdO4jjh-Neo